



30th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM2021) 15-18 June 2021, Athens, Greece.

FlexRQC: Model for a Flexible Robot-Driven Quality Control Station

A. González Tomé, I Irigoien Ceberio, U. Ayala, J. A. Agirre, N. Arana-Arexolaleiba*

Mondragon University - Faculty of Engineering Goiru, 2; 20500 Arrasate - Mondragón (Spain)

* Corresponding author. Tel.: +(34) 943794700. E-mail address: narana@mondragon.edu

Abstract

As the investment on a dedicated quality control stations is not desirable for limited production batches. In general, those systems result in very optimised systems and the lack of flexibility since they are designed for an ad-hoc production. To provide a solution for those cases, a new model to design a flexible quality inspection system is proposed. This paper introduces FlexRQC (Flexible Robotic Quality Control) a model for characterising flexible robot-driven quality control stations. FlexRQC is divided into two domains: The Quality Control Station Domain (QCSD) and the Model Under Inspection Domains (MUID). FlexRQC takes advantage of 3D CAD systems to get spacial information on the quality control station and the quality requirement. The flexibility of the model has been successfully tested in two quality control station setups and various solid rigid objects.

© 2020 The Authors. Published by Elsevier Ltd.

This is an open access article under the CC BY-NC-ND license (<https://creativecommons.org/licenses/by-nc-nd/4.0/>)

Peer-review under responsibility of the scientific committee of the FAIM 2021.

Keywords: Robotics; Flexible Quality Inspection; Programming;

1. Introduction

Current quality control systems used for the quality control assurance of the fabrication processes of solid objects are most commonly dedicated systems, that target a single model or a series of similar ones. This degree of specialisation of the current Quality Control Stations (QCS) allows them to achieve high levels of efficiency, and thus, are desirable for mass production. However, in the case of those product batches that are manufactured on-demand, usually for a limited period, the requirements continuously change. In such productions scenarios, dedicated QCSs are not viable investments. However, quality assurance is still desired [1]. To meet the requirement of continuously changing demand, QCSs that are flexible and can inspect different types of objects are required.

In [2], the flexibility provided by robots as central agents of QCSs has been demonstrated. FlexRQC will also consider the robot as a prominent player and will be in charge of offering flexibility. The structure and tools that this QCS will contain are uncertain as different Models Under Inspection (MUI) will require different inspection technologies. To be able to define

unequivocally the elements involved in the flexible QCS, a description conceptual framework must be specified. This sort of frames, referred to as ontologies, are well known and have been developed for many domains of knowledge. Besides, they offer development process [3], tools [4] and validation methods [5].

This work aims to develop a model for flexible robot-driven QCSs systems, named FlexRQC (Flexible Robotic Quality Control). As a guideline for the robotic part of the model, the IEEE CORA [6] ontology was used. Unlike others [7], [8], the model proposed in this work extends previous models enabling to create a technology-independent, flexible inspection system. The FlexRQC model serves to ease the definition and implementation of the inspection procedure of MUIs. Tools capturing the information from 3D CAD software and transferring into the FlexRQC model has also been developed. This user-oriented approach allows the model's implementation to be usable.

2351-9789 © 2020 The Authors. Published by Elsevier Ltd.

This is an open access article under the CC BY-NC-ND license (<https://creativecommons.org/licenses/by-nc-nd/4.0/>)

Peer-review under responsibility of the scientific committee of the FAIM 2021.

10.1016/j.promfg.2020.10.013

2. FLEXRQC MODEL

This section describes the FlexRQC model. The FlexRQC model starts by defining the two main elements present in a quality control scenario. These are (i) the technologies and supporting hardware that will run the inspection and (ii) the target of the inspection, the QCS and the MUI, respectively.

As a QCS example, in Fig. 1 a laboratory prototype containing five inspection technologies can be seen. In this prototype, there are two 2D inspection technologies, both integrated into the same post. The first one uses a direct-light and the second one a back-light. Besides, it contains three 3D inspection systems: a 3D laser scan, a deflectometry [9][10] and a photometric stereo.

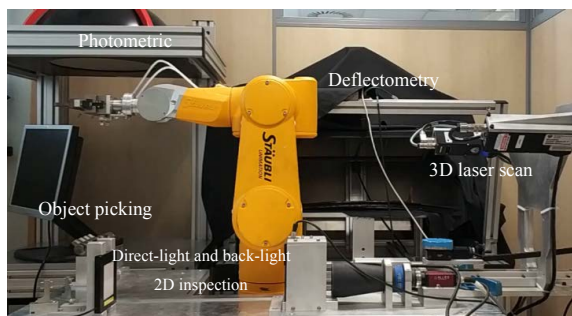


Fig. 1: A lab. prototype with multiple inspection tech

QCS and MUI will now be further described with the aid of a UML diagram (Fig. 2). The QCS can be defined as a cell or installation that is responsible for performing the inspection. Being this the case of a robot-driven system, an essential

object to be inspected from the picking position (Fig. 1-Fig. 3), though the different QCSs and take back to the final place position (it can be the picking position). The QCS is also composed of posts. A post is a region of the QCS that may have different functionalities. These posts can be a subsystem that performs quality control or a picking area. Additionally, the MUI is defined as the theoretical 3D object (CAD model) of the object to be inspected. It also contains the quality criterion of the model.

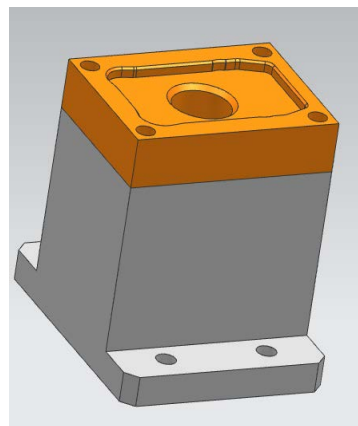
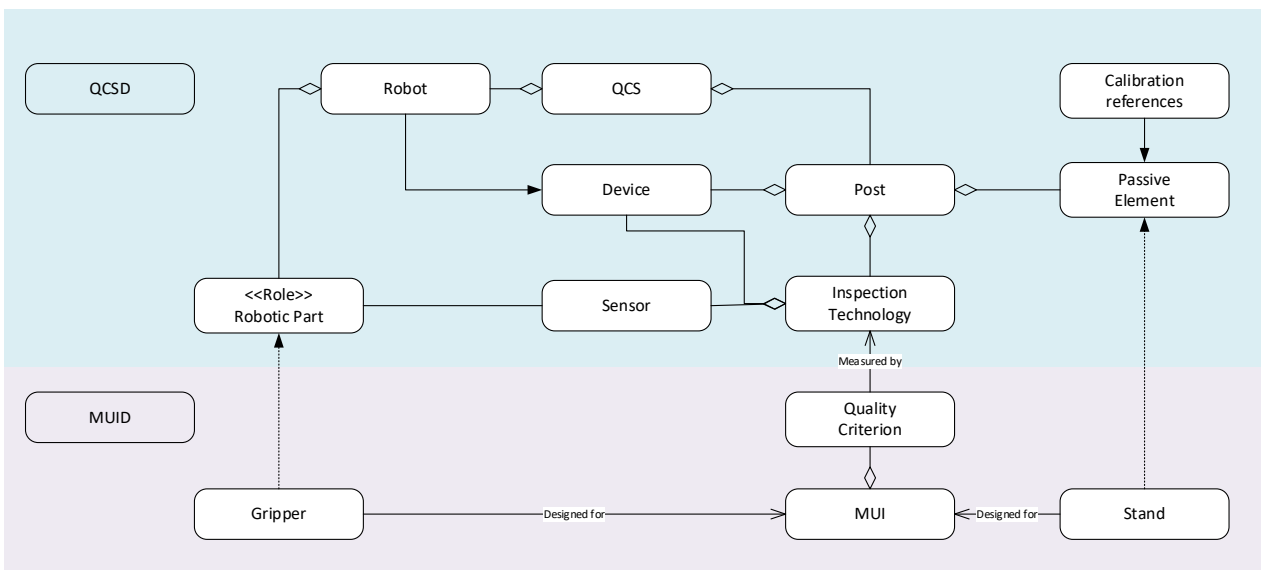


Fig. 3. The picking post

The main elements of the Quality Control System Domain (QCS) are the following:

- **Robot:** The robot is the central element of the system. It is the agent responsible for making the whole system operate in a unified manner. It does by defining a task-based behaviour (pick, place, face the object to the camera, move it from one post to the other, etc.) enabling to inspect with different



element of the QCS is the robot itself. The robot is a task-based agent that operates the QCS. The main robot task is to move the

available technologies.

Fig. 2: A simplified UML model of the FlexRQC model

Usually, this means that the robot will be carrying the object around the QCS from post to post. Grippers are, as defined in the IEEE CORA [6], robotic parts, elements attached to the robot.

- **Post:** Posts are regions of the QCS. Typically one of the next three functions may be performed in a post: picking, disposal or inspection of the MUIs. Different services can be completed in each posts depending on the available devices. Those devices can be active elements such as cameras, laser scan, lighting system, robot gripper exchangers or linear or rotatory stages (i.e. to scan the objects). However, post can also contain only passive elements; the most common ones are pick and place containers (Fig. 3) or robot calibration references. The inspection posts usually mixe both active and passive elements. The post also contains inspection technologies. They are sensors and devices that perform any different data acquisition ranging from 3D or 2D cameras to simple ones (touch sensor). As an example, an inspection post may serve to do a back-light inspection using a 2D camera together with the back-light), a rotatory plate for positioning (active element) and a stand to place the object (passive element).

On the Model Under Inspection Domain (**MUID**), the main element is the quality criterion.

- **Quality Criterion:** Quality Criterion define a characteristic that is to be measured for each MUI together with its ideal values and tolerances. Quality Criterion must thus represent a property of the object that can be both measured and compared, ranging from dimensional measurements (i.e. a width - Fig. 11, a radius - Fig. 12, a diameter and a centre - Fig. 4) to surface inspection, such as aesthetic imperfections.



Fig. 4: Center and diameter

Additionally, several relations are required between the QCS and the MUI in the model to guarantee successful quality control of the object. The first one originates from the need of the QCS to be adapted to each MUI. The unforeseeable geometries of the MUIs make the design of the handling elements dependent on the object to be inspected. In such a case, the grippers and stands, have to be designed for a specific MUI, which implies that the QCS has to be designed in a way that allows the interchange of these elements depending on the selected MUI. Apart from this, the quality criterion and inspection technologies are also closely related. While the quality criterion described in the MUI define what needs to be measured, the inspection technologies are the resources to

measure the quality criterion. As a result of this relation, the quality criterion has to be linked to a specific inspection technology with the capacity to measure them.

A robot-driven approach need involves the precise guidance of the movements of the robot across the QCS. To do so, the exact pose of all the elements in the QCS must be obtained, and the spatial transformation tree must be established. This transformation tree helps to achieve the relation among all pose of interest in the QCS. Some examples of poses of interest are: object picks and places poses, image capturing poses, and laser scanning pose.

For instance, in order to obtain the pick location of a stand, the posterior chain will be used: from the robot to the post, from the post to the stand's place and from there to the picking location in the stand. The tool positions, positions relative to the flange of the robot used for commanding it, are also calculated similarly, mainly depending on the quality criterion' information. The model is designed taking this nesting into account to increase modularity and, together with that, flexibility.

3. IMPLEMENTATION

The FlexRQC model was used to produce a prototype for medium-sized (3-30cm) solid object inspection. The system had to fulfil a set of requirements taking into account the use that the system will have.

- The data contained by the model must be stored and exchanged. For this XML was chosen as information representation format. As it is a widely established standard with plenty of serialisation options for almost any computing language. It also allows representing hierarchical information in human-readable form easing development and usability.

- The data contained by the model must be generated quickly. Due to the nature of the information that has to be recorded (Cartesian 3D positions mainly) the direct generation of an XML file by human user results in a too tedious process. This process was solved by creating a requirement specification module on top of a 3D CAD file.

- The responsibility of each user must remain within his domain of expertise. This means that the user responsible for the annotation of the MUI does not have to have more than basic knowledge of robotics or vision systems. One user is responsible for identifying and targeting the critical characteristics of each MUI, while another user defines the setup of the QCS, including the inspection technologies available. The final assignation between quality criterion and inspection technologies is to be performed using a smart expert system that takes into account the characteristics of each quality criterion and the capabilities of the available technologies.

- The definition of the QCS is MUI independent. This allows multiple MUIs to be applied in a single QCS. This requirement was complied by separating elements by dependency and not physical hierarchy. This results in the division of the information into two separate XMLs: one for the QCSD (QCS Description of the cell) and the other for MUID (MUI Specification of the quality criterion to be measured) files.

- QCS: The QCS is the file containing the information on the QCS, and it is not MUI dependant. It includes the information of all the hardware present on the QCS and their configurations. However, it does not contains those elements that depend on the MUI, such as robotic grippers and the stands.

- MUID: Complementary to the QCS file, the MUID file includes the gold standard values and tolerance ranges of the quality criterion of the MUI. The declarations of the quality criterion should be ultimately inspection technology-independent. However, this is not entirely true for the implementation that was carried out, and some leakage was allowed to ease the inspection technology assignment process. It also contains the elements that were excluded from the QCS like the grippers and the stands.

The system has been tested in a QCS with the following characteristics:

- Back-light: An Opto Engineering Telecentric Lens TC2MHR240-C (200x170mm) and a 9.2Mpix Camera camera and an area back-light.
- 3D images. Gocator2340 laser scanners.
- Robot: Stäubli TX90

A 3D CAD file was generated, representing the set up of the system using the Siemens NX application. This CAD file is the source to create the QCS file. Complementary to this, another CAD file was created from who the MUID file was generated. This CAD file contained the MUI, and its dependent elements (grippers and stands). A plugin for the Siemens NX application was created to extract information from the CAD files. This plugin was developed in C#/.NET using the NXOpen library, which NX provides for plugin creation. The plugin allowed to create the inspection requirement process and to export it to QCS's (Fig. 6) and MUID's (Fig. 7) XML files.

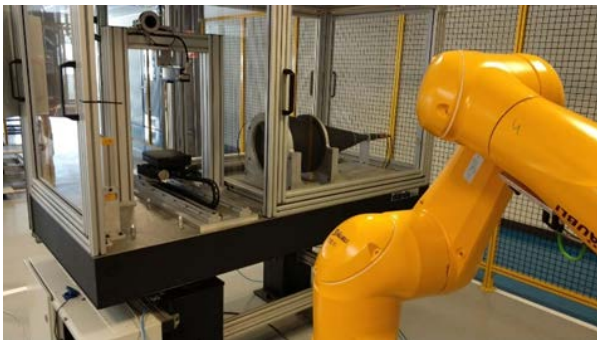


Fig. 5: Full-size prototype

The obtained data was then fed to the software module in charge of executing the inspections process on a desktop PC. This module, developed in C#/.NET, is also in charge of controlling all the devices. The robot for safety compliance is connected through a PLC system. The robot is controlled through a master-slave command-based communication over TCP/IP protocol, where the robot is the slave and the PC the master. For the analysis of the obtained images, HALCON 12 image processing library was used.

3.1. XML

The QCS and MUID are the main elements that are fed into the execution software. As previously mentioned, these elements were implemented as XML files. To illustrate the usage and functioning of this XML files an example will be provided.

In a usual setup, a single QCS is used to define the capabilities of the inspection cell while different MUIDs will be used which determine both different MUIs and routines for each MUI.

As an example, a definition of a QCS containing two inspection technologies (2D back-light and 3D laser sensor) is provided in Fig. 6. Note that this XML extract is a simplified version of the original one. In this extract, it can be seen how the cell is composed of the inspection posts and which are resolved into a software implementation due to their name (Post_GOCATOR and Post_RGB). The passive element serves as position referencing for the pick and place.

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <QCS>
3   <Post>
4     <Post Nombre="Post_GOCATOR">
5       <Devices>
6         <Device Nombre="GOCATOR"/>
7       </Devices>
8     <Devices>
9       <PassiveElement />
10    </Devices>
11    <PosSafety Name="POS_SAFETY_Post_GOCATOR"/>
12  </Post>
13  <Post Nombre="Post_RGB">
14    <Devices>
15      <Device Nombre="Device_BACKLIGHT"/>
16    </Devices>
17    <Utiles />
18    <PosSafety Name="POS_SAFETY_Device_RGB"/>
19  </Post>
20 </QCS>

```

Fig. 6: XML of QCS

As an example of a MUID's XML, an object with a quality criterion measured by back-light imaging and another measured by laser profiling is given. The simplified XML extract can be found in Fig. 7. The quality criteria defined in this example are the ones used later in the paper for validation of the system, and examples can be seen in Fig. 9 and Fig. 10. This is defined as the "QualityCriterion" elements in the XML. Where each of them is surrounded by a capture element "VIEW". Two captures are present, "VIEW1_" is defined as a capture meant for the 3D laser profiler ("VIEW3D" tag) while "VIEW2_" is directed to the 2D camera (by default). It can be seen in the XML how the tolerances are defined for the "Width" quality criterion and how the two points used for referencing the measurement points are assigned. These are transformed into image coordinates during execution and are shown in Fig. 9 as red dots. The quality criterion defined in Fig. 7 illustrates that the width should be 43mm \pm 0.1mm.

```

1 <MUI>
2 <Object>
3 <Views>
4 <View xsi:type="View3D">
5 <Name>VIEW1_</Name>
6 <QualityCriteria>
7 <QualityCriterion xsi:type="Heigh" Name="PC_Heigh_1">
8 </QualityCriterion>
9 </QualityCriteria>
10 </View>
11 <View>
12 <Name>VIEW2_</Name>
13 <QualityCriteria>
14 <QualityCriterion xsi:type="Width" Name="PC_Width_1">
15 <ExpectedDim xsi:type="Distance">
16 <TolerancePlus>0.1</TolerancePlus>
17 <ToleranceMinus>-0.1</ToleranceMinus>
18 <Range>43</Range>
19 </ExpectedDim>
20 <GripPosition>GRIP_POS_OB1</GripPosition>
21 <Point1/>
22 <Point2/>
23 </QualityCriterion>
24 </QualityCriteria>
25 </View>
26 </Views>
27 </Object>
28 <Gripper/>
29 <Stand />
30 </MUI>
    
```

Fig. 7: XML of MUID

4. RESULTS

In terms of the adaptability of the system, two different prototypes have been created and operated. The first was a laboratory prototype which worked with various inspection technologies with the robot (Fig. 1). The second was the industrial prototype (Fig. 5) which included two inspection technologies (2D back-light and 3D laser sensor) with an intermediate device. The intermediate device consisted of stand-mounted on a combination of linear and rotatory actuators. This intermediate device performed the placing of the object to be inspected in front of the cameras. The flexibility of the system was demonstrated using the same MUIs' model both cells, each one with each own specific QCSO files.

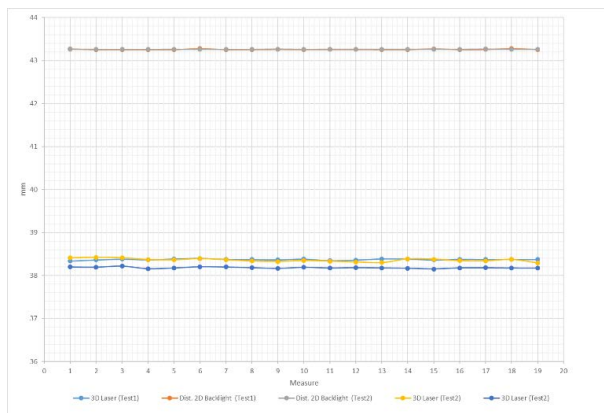


Fig. 8: Qualitative tests (2D Backlight and 3D laser)

In the full-scale prototype, a preliminary test was done using both the 2D back-light and the 3D laser sensors to assess the precision of the system. The test has consisted of a series of

measurement. Two sets in the case of the 2D back-light and three in the case of the 3D laser.



Fig. 9: An image of distance measurement using back-light on object 1

Each measurement process included the next steps: First, the object is taken from the picking place. After, the robot places the part in front of the back-light camera and then below the laser scanner. Finally, the object was returned to the initial position. The robot always starts and finish all the movement in its home position. Without any human intervention, the measurement process started again.

The first 19 measurement the 3D laser measurement standard deviation is lower than 0.04mm and 0.008mm in the case of the 2D back-light. It was observed that the initial position of the object affects the results.

Fig. 9 shows an image of the distance measurement with the 2D back-light sensor and Fig. 10 the profile of the 3D laser scanner.

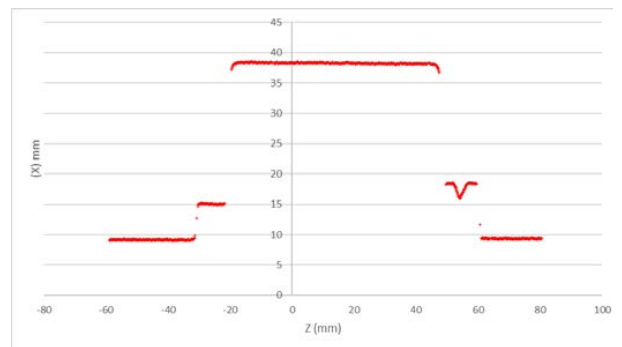


Fig. 10: 3D Laser profile on object 1

Besides, three different MUIs have been tested on the system. The centre and diameter (Fig. 4), the with between two lines (Fig. 11) and a radius (Fig. 12) were measured.

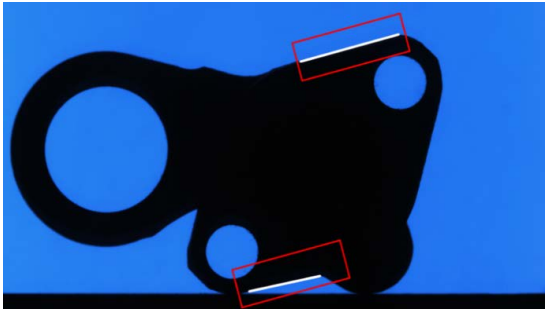


Fig. 11: Width (object 2)

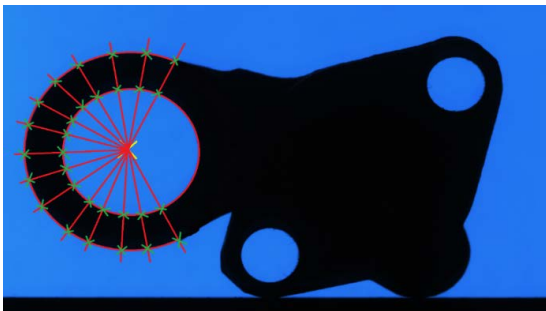


Fig. 12: Radius (object 2)

The 3D laser scanning was tested in the laboratory prototype. In this case, instead of taken only one profile, a complete scanning was tested. Fig. 13 shows the cloud point of the object 2 without any filtering, where even the robot's gripper can be observed. And Fig. 14 shows the 3D laser scanning of a larger object in the same laboratory prototype. In this case, the cloud point filtering algorithm has been applied.

The definition of the MUIs file was quickly performed on top of CAD models and the manufacturing of the required hardware, grippers and stands, was also be delivered promptly with fast prototyping systems (3D printing).

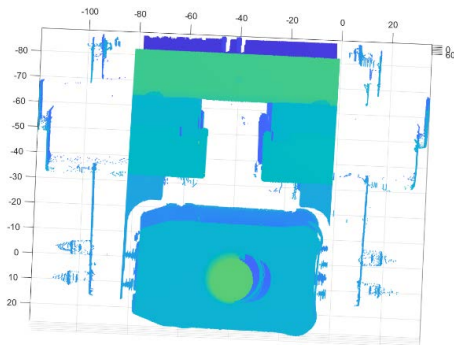


Fig. 13: 3D laser scan of the object 1 on lab prototype

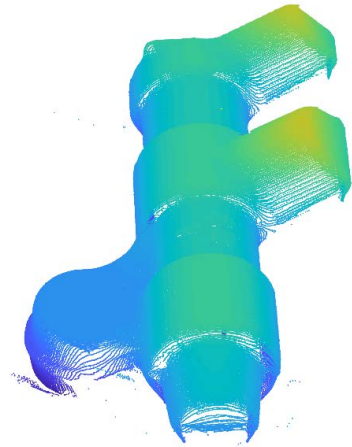


Fig. 14: 3D laser scan of the object 3 on the lab. prototype

5. CONCLUSION

This work is a first attempt to create a model that joins both the robotic and quality control domains with the intent of creating a flexible QCS that will allow for proper automatic quality assessment of objects. The proposed model flexibility was tested in two different prototypes, three objects and two inspection technologies (3D laser and back-light).

The operation of the requirement definition was taught to a user. The user was not involved in its development. The user was able to independently generate MUIs XML files just with rough knowledge on the inspection technologies involved

The software is also oriented towards the introduction of new inspection technologies. Several technologies have already been introduced during the research process, 3D surface analysis, 3D volumetric inspection, profile analysis, silhouette inspection and basic OCR for object identification. This variety together with the possibility of rapidly integrating new technologies gives the system a high level of adaptability.

The increasing performance of state-of-the-art industrial robots and machine vision systems and algorithms enable high precision measurement. Although this system is currently undergoing a testing process, preliminary results show high precision on the measurements made with a standard deviation below 0.04mm.

The main limitation in the implementation is that the XML files are serialised and deserialised directly into C# classes using .NET's default implementation. This means that the generated XML is C# dependent. In future works, the programming language dependency should be avoided, and an XML validation schema should be developed.

Acknowledgements

This work has been developed by the Intelligent Systems for Industrial Systems group partially supported by the Department of Education, Language Policy and Culture of the Basque Government and Idefix (PUE 2018-06) research project. The

authors would also like to thank the collaboration partner Ekide S.L. for their contributions.

References

- [1] P. Zheng, H. wang, Z. Sang, R. Y. Zhong, Y. Liu, C. Liu, K. Mubarak, S. Yu, and X. Xu, "Smart manufacturing systems for Industry 4.0: Conceptual framework, scenarios, and future perspectives," *Frontiers of Mechanical Engineering*, vol. 13, no. 2, pp. 137–150, Jun. 2018. [Online]. Available: <http://link.springer.com/10.1007/s11465-018-0499-5>
- [2] M. Kefer and J. Tian, "An intelligent robot for flexible quality inspection," in *2016 IEEE International Conference on Information and Automation (ICIA)*. Ningbo, China: IEEE, Aug. 2016, pp. 80–84. [Online]. Available: <http://ieeexplore.ieee.org/document/7831800/>
- [3] M. Fernández-López, A. Gómez-Pérez, and N. Juristo, "Methodology: from ontological art towards ontological engineering," 1997.
- [4] A. Diego, V.-C. Cristina, O. Francisco, P. Juan, and Á. Bárbara, "V3cmm: A 3-view component meta-model for model-driven robotic software development," *Journal of Software Engineering in Robotics*, vol. 1, no. 1, pp. 3–17, 2010.
- [5] N. Guarino and C. Welty, "Evaluating ontological decisions with ontoclean," *Communications of the ACM*, vol. 45, no. 2, pp. 61–65, 2002.
- [6] E. Prestes, J. L. Carbonera, S. Rama Fiorini, V. A. M. Jorge, M. Abel, R. Madhavan, A. Locoro, P. Goncalves, M. E. Barreto, M. Habib, A. Chibani, S. Gérard, Y. Amirat, and C. Schlenoff, "Towards a core ontology for robotics and automation," *Robotics and Autonomous Systems*, vol. 61, no. 11, pp. 1193–1204, Nov. 2013. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0921889013000596>
- [7] S. R. Fiorini, J. L. Carbonera, P. Gonçalves, V. A. Jorge, V. F. Rey, T. Haidegger, M. Abel, S. A. Redfield, S. Balakirsky, V. Ragavan, H. Li, C. Schlenoff, and E. Prestes, "Extensions to the core ontology for robotics and automation," *Robotics and Computer-Integrated Manufacturing*, vol. 33, pp. 3–11, Jun. 2015. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0736584514000659>
- [8] R. Farinha and P. Gonçalves, "Knowledge-based robotic system, towards ontology-driven pick and place tasks," vol. 2016, pp. 152–157, Jan. 2016.
- [9] Daniel Maestro-Watson, Julen Balzategui, Luka Eciolaza, Nestor Arana-Arexolaleiba, "Deflectometric data segmentation for surface inspection: a fully convolutional neural network approach," *J. Electron. Imag.* 29(4) 041007 (4 February 2020) <https://doi.org/10.1117/1.JEI.29.4.041007>
- [10] D. Maestro-Watson, J. Balzategui, L. Eciolaza, and N. Arana-Arexolaleiba. Deflectometric data segmentation based on fully convolutional neural networks. In *SPIE International Conference on Quality Control by Artificial Vision (QCAV)*, volume 11172, 2019. [Best Paper Award]